John von Neumann Institute for Computing

Data Structures and Mesh Processing in Parallel CFD Project GIMM

B.N. Chetverushkin, V.A. Gasilov, S.V. Polyakov,
M.V. Iakobovski, E.L. Kartasheva, A.S. Boldarev,
A.S. Minkin

published in

Parallel Computing:
Current & Future Issues of High-End Computing,
Proceedings of the International Conference ParCo 2005,
G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **33**, ISBN 3-00-017352-8, pp. 351-358, 2006.

http://www.fz-juelich.de/nic-series/volume33

Data Structures and Mesh Processing in Parallel CFD Project GIMM

B. N. Chetverushkin[a], V. A. Gasilov[a], S. V. Polyakov[a], M. V. Iakobovski[a], E. L. Kartasheva[a], A. S. Boldarev[a], A. S. Minkin[a]

[a]Institute for Mathematical Modelling, Russian Ac.Sci., 4-A, Miusskaya Sq., Moscow 125047

Key words: Computational fluid dynamics, three-dimensional evolutionary viscid flow, parallel algorithms, distributed computations.

The study is done under the auspices of Russian Ac. Sci. (The State contract No. 10002-251/OMH-03/026-023/240603-806) and RFBR (Project No. 04-01-08024-OFI-A).

1. Introduction

Rapid development of networks caused the growth of interest to heterogeneous systems applications in remote computations resulting thereby in the progress of special software. In this paper we present a CFD project GIMM carried out in the Institute for Mathematical Modeling, RAS. GIMM pursues the goal of joining up new results in numerical analysis with latest achievements in creation of network facilities. Special attention is paid to the reasonable management of multiprocessor systems (MPS) with distributed memory. Design of new parallel algorithms and proper programming is considered as having paramount importance in effective exploitation of parallel systems. Design of new numerical methods is implemented in close correlation with the development of algorithms for parallel computing. The use of high-dimension irregular meshes permits to approximate accurately realistic 3D geometry of the simulated objects. Processing of irregular data structures is a complicated and very time-consuming numerical work. New effective algorithms of domain decomposition over processors taking into account numerical expenses in each point are developed that provides the good load balancing and minimize data exchange.

2. General description

The current version of GIMM components includes:

- CAD compatible data structures,
- Surface/solid grid generation tools,
- Libraries of a numerical package kernel,
- Problem-oriented application software for PC and parallel systems,
- Server tools for data storage, visualization, pre- and postprocessor tools for PC and parallel systems,
- Client tools for data storage, visualization, pre- and postprocessor tools for PC,
- Server control tools for processing the user tasks by parallel system,
- Client interface for PC.

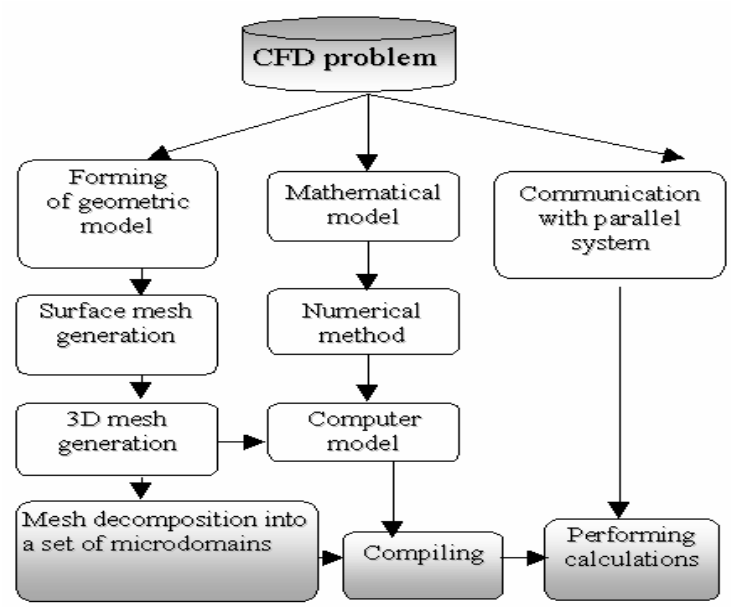

Figure 1. GIMM program manager.

The work of software components is supervised by the GIMM program manager. The first program manager version works under Windows and ensures the interactive user work with remote computing resources and computer systems possessing both distributed and shared memory. The server modules of the GIMM package are developed as Linux applications. The MPI is used for task start and interprocessor communications during the run. In case when the mixed type parallel systems are used for calculations, e. g. when the two- or four-processor nodes are incorporated into the computational net, the simulation can be implemented by means of both MPI and OpenMP utilities. To this end the work at the processors' level is done via algorithms using the common memory while at the nodes' level the appropriate algorithms use MPI utilities and distributed memory.

The program manager supports operations usual for CFD studies:

1. Preprocessing I, i. e. creation of a geometry model and setting of physical data as well as initial/boundary conditions.
2. Preprocessing II, or mesh generation and creation of a computational model equipped by problem attributes.
3. Formation of algebraic equations approximating the governing system of differential fluid mechanics equations.
4. Support of the task starting operations and run control functions.
5. Postprocessing, or data analysis and visualization in the remote operational mode. The program manager structure and main functions are shown on Fig. 1.

3. Numerical methods

At present in GIMM the following models are available:

- Model 1: 3D Navier-Stokes system for compressible heat-conductive flow;
- Model 2: 3D Navier-Stokes system for incompressible flow;
- Model 3: 3D single-phase nonlinear flow in porous media.

The governing systems are approximated by means of unstructured tetrahedral meshes and conjoint systems of finite volumes. The mixed finite volume (FV) — finite element (FE) approximations provide sufficiently high accuracy: the approximations to convective terms are done in terms of FV technique, and dissipative terms are approximated by simplex FE representations of dependent variables. For treatment of convective fluxes we use high-resolution TVD schemes of Roe, Osher and Van-Leer [1]. We also use the Chetverushkin kinetically-consistent method [2] and TVD Lax-Friedrichs schemes with intermediate solution reconstruction which we extended for the case of 3D unstructured meshes [3]. The time-marching algorithm is implemented as an explicit 2-nd order predictor-corrector.

3.1. Geometric models and meshing

Preprocessing related to geometric properties of a studied object usually is implemented by means of some CAD system. A set of universal program tools is developed for input and acquisition of geometrical and physical data related to a 3D problem with mixed initial and boundary conditions. Simultaneously these tools are used for storage and treatment of a discrete computational model. Our data model takes its origin in finite element technique and provides a description of geometric and topological properties for computational domains as well as for numerical meshes. The geometry of any computational model is formed of parametric surfaces expressed in the terms of rational B-splines (NURBS). Any 3D geometric complex is represented via sets of elements: 0-order elements = "nodes", 1-order elements = "edges", 2-order elements = "faces", 3-order elements = "cells".

A topology complex describes the relations between numbered data sets irrelative of their geometric nature. Typical 3D meshes consists of a large number of cells. Therefore only a few incidence relations are stored permanently and other relations are calculated every time they are needed.

The main topological relation between pairs of cells in a cellular complex is "to be a face". It is also named as a boundary relation. A cell d is a face of a cell e if d belongs to the closure of e. If $d \neq e$, then d is said to be a proper face of e. Other relations can be defined on the base of this one. Two cells e and d are incident if d is a proper face of e, or e is a proper face of d. Two cells e and d are adjacent if there exists another cell in the complex which is a proper face of both e and d. Thus the incidence relations serve for elements of different dimensions, and while adjacency relations serve for elements having the same dimension.

We suppose that computational mesh can change in the process of calculations. Data assigned to mesh elements are not predefined and can vary from one application to another so we introduced the conception of a numerated set. We use such sets for representation of collection of mesh elements. We imply that all mesh elements are enumerated and each of them has its unique number. So, having the number of an element, it is possible to find all the data assigned to this element. Numeration allows to implement changes in all data structures when some elements are added or deleted.

Evidently realistic results in 3D simulations can be achieved starting with meshes consisting of about $10^6 \div 10^7$ nodes or more. For such meshes the proper amount of a processed information is so large that it is necessary to develop parallel mesh generation algorithms. For applications in parallel computing we developed a technique based on domain decomposition with the following mesh generation by means of combined octree — modified advancing front algorithms [4] permitting to generate cells of prescribed shapes and sizes. The octree technique is applied for mesh generation

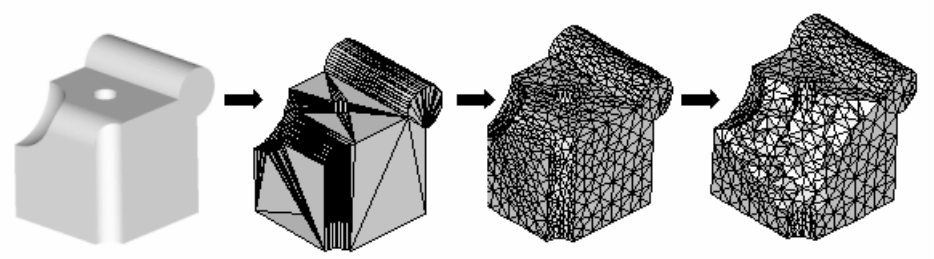

Figure 2. A scheme of mesh generation in a solid.

in the interior of any subdomain while the advancing front algorithm works in the near-boundary regions. Some optimizing procedures are developed for getting meshes of the higher quality. The initial front is formed by surface triangulation (see Fig. 2). The technique of multicolouring allows to implement parallel meshing in subdomains in such sequence that avoids the data exchange between adjacent regions and ensures fully independent meshing of subdomains.

3.2. The data treatment for the high-dimension meshes

Some problems are to be solved for the effective use of high-dimension meshes. The one is the correct load balancing whish is a very important factor strongly affecting a rate of parallel computations.

Another problem providing the evident difficulties in large-scale 3D simulations is the lack of acceptable visualization tools. For a typical mesh consisting of $10^6 \div 10^7$ nodes the whole amount of data exceeds the memory resources of a personal computer.

Let's consider the main aspects of computational technologies concerning the large volume data treatment created under the GIMM project.

The load balancing is done by means of the rational decomposition of the computational mesh via the constructing of a multilevel graph system. Due to this method mesh is represented as an ensemble of coherent subdomains. The initial graph is formed by pairs of neighbor mesh vertices. In turn, these pairs are combined into the higher level pairs. Thus the multilevel graph decomposition is formed successively, while every new graph possesses a simpler structure than the previous one. As the result of this work the computational mesh is represented as a set of connected subdomains. This procedure repeats until the graph of a proper size is obtained which can be easily divided into the required number of macrodomains. At the farther stage of the run these macrodomains are distributed over the set of processors. The macrodomains serve also for estimations of the numerical work. During the macrodomain creation performance it is possible to take into account not only the mesh topology, but an a priori estimated numerical complexity of calculations per every mesh node. Thus the resulted load distribution can be made very close to the optimum.

The data storage in the GIMM code is organized via the data distribution over the multiprocessor system itself. This technology is supported by specially developed hierarchical distributed file system (HDFS). It is based on the client-server technology in the multiprocessor variant. In accordance with this technology some number of specially appointed processors are used for communications through the entire disc space of the massive parallel system (MPS) including as local (processor self memory) as external (RAID server) devices. The other processors implementing the run are the clients connected to definite servers. The read/write operations corresponding to the distributed files

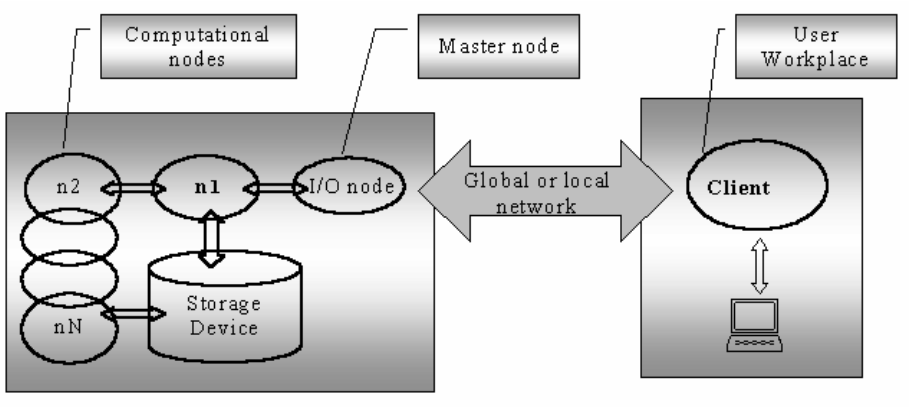

Figure 3. RemoteViewer: application of a client-server technology

are performed via the system of demands addressed to the HDFS servers. The cash memory using leads to somewhat information doubling which is almost inevitable. Therefore the main idea of organizing the HDFS system is to minimize the doubling of the information stored on the individual discs and simultaneously to maintain the high exchange rate and information integrity. In addition the HDFS is used for data compressing necessary for data storage and further treatment at the postprocessing stage. The convenient tools for data visualization and postprocessing are of primary importance in 3D simulations of complex FD problems. Here one should take into consideration the fact that typical volume of the 3D results obtained with a mesh of $\sim 10^6 - 10^7$ exceeds the storage capability of an individual PC. Besides, the data exchange through the net connecting PC with MPS is restricted by other operations related to multiprocessor computations.

To mitigate this problem we developed the distributed visualization system which is especially suitable for treatment of large data volumes. The GIMM visualization system RV — "Remote Viewer" is constructed according to a Client/Server model. It works in close interaction with HDFS. This construction allows to implement the most part of visualization process by supercomputer and then to transfer the compressed information to a user (see Fig. 3).

The final image is formed at the user's workplace and it is possible to use modern multimedia hardware (helmets, stereo glasses, three-dimensional manipulators etc.) for better image perception.

4. Numerical results

At modern stage of the GIMM development we performed hydrodynamic test studies pursuing the two main goals:

- approbation of the algorithms and numerical techniques (accuracy, efficiency, etc.);
- study of the effectiveness of parallel processing different distributed computing systems architecture.

The package working ability was examined through a number of test studies. Here we present a brief description of results pertinent to such famous benchmark as the problem of viscid low-Mach ($\mathrm{M} \sim 0.1$) and low-Reynolds ($\mathrm{Re} \sim 25$) flow around a sphere. A numerical grid was constructed so to condense in the sphere vicinity, and the volumetric ratio of largest cells to smallest ones was ~ 100. The total mesh consisted of 2 356 196 nodes and 14 018 176 tetrahedrons. Note that for

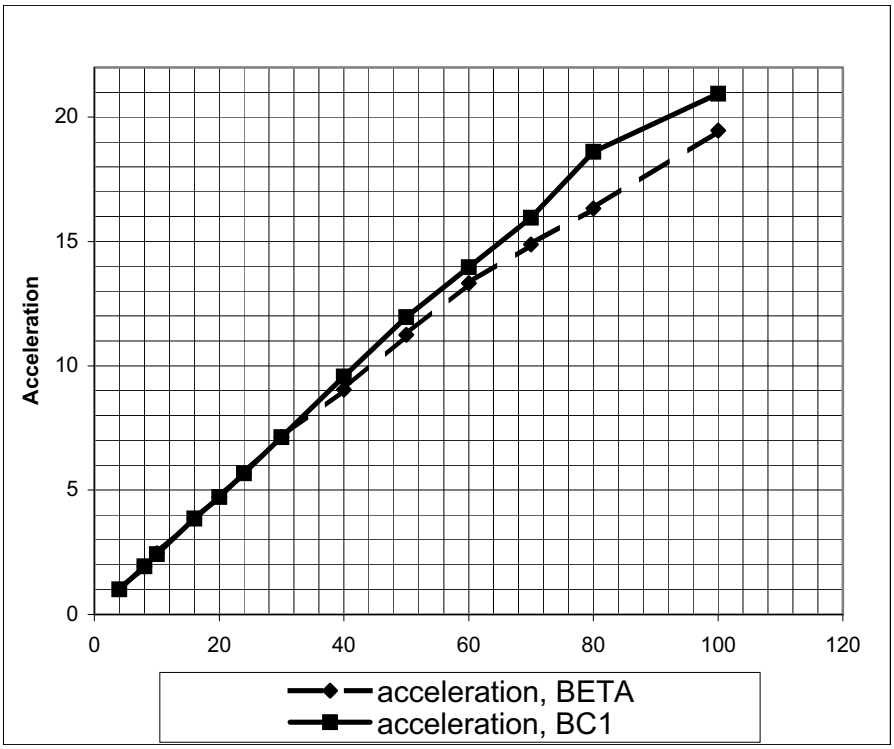

Figure 4. Acceleration rates for BETA and BC1 parallel systems.

the given mesh parameters the calculations with only a single-processor computer are almost out of perspective as we consider reasonable time of getting the result.

The steady-state calculations were done by the time-marshing procedure. For the test studies we used the two parallel systems operating in the Joint Supercomputer Center (Russian Ac. Sci.): MVS-1000 (equipped by Dec Alpha 21167 processors) and MVS-5000 (equipped by IBM Power P4 processors). The acceleration according to increased number of the used processors was estimated by the formula $S_N = T_4/T_N$, where N is the number of processors, T_N is the computing time for these processors. The effectiveness was estimated by the formula $E_N = (S_N \cdot 4/N) \cdot 100\% = (S_N/(N/4)) \cdot 100\%$. One may see from the figure 4 that maximum effectiveness for MVS-1000M was 98.3% when $N = 10$, and the minimum effectiveness was 77.8% when $N = 100$. These results demonstrate that the algorithm is efficient even when using strongly irregular tetrahedral meshes.

5. Conclusion

Dealing with the standard industrial prgram tools the user may encounter the restrictions of his activity caused to some extent by the stiff program structure or by the set of standard models and methods. Being reasonable in CAD-CAE project design, such restrictions fall into contradiction with the concept of a research code which is often aimed at studies of new models and algorithms. Considering the application aspects of high-performance calculations one may conclude that the industrial codes yet can not use perfectly all the resources which can be given by modern massive computational systems. Keeping in mind these points the GIMM team concentrated efforts on the development of open code with predominant applications in the area of parallel and distributive computations. The main goal pursued at the first stage of the project development was the develop-

ment of versatile algorithms suitable for various applications in CFD studies. This was the principal motivation for the development of numerical algorithms using unstructured meshes and their adaptation to calculations with various multiprocessor systems. The modern version of the GIMM code is aimed at numerical simulation of evolutionary multiscale 3D hydrodynamic phenomena. It combines as traditional as new numerical technologies which were incorporated into the code with the primary aim to have a universal simulation tool. GIMM provides a possibility of massive parallel computations with $10^8 \div 10^9$ mesh nodes and by the order of 10^3 processors involved. The main feature of GIMM package is the comprehensive use of parallel programming at all stages of a problem solution, i. e. from geometry modeling till postprocessing. Some numerical tools which are already incorporated into GIMM were developed 3–5 years ago and passed through comprehensive approbation in IMM RAS. The created code is highly effective compared to the traditional PC-oriented tools and allows 10 to 100 times reduction of the period necessary for the numerical study of as fundamental as applied problems.

References

[1] D. Kroner: Numerical schemes for conservation laws. B. G. Teubner Publ., Stutgart, Leipzig. 2000.

[2] B. N. Chetverushkin: Kinetic schemes and quasigasdynamic system. MAKS Press, Moscow. 2004.

[3] V. A. Gasilov and S. V. D'yachenko: Quasimonotonous 2D MHD scheme for unstructured meshes. Mathematical Modeling: modern methods and applications. Moscow, Janus-K. 2004. P. 108–125.3

[4] P. J. Frey and P. L. George: Mesh generation. Hermes Sci. Publ., Oxford, UK. 2000.

[5] M. Yakobovski, S. Boldyrev, and S. Sukov: Big Unstructured mesh Processing on Multiprocessor Computer Systems. In: Parallel Computational Fluid Dynamics - Advanced numerical methods, Software and Applications / B.Chetverushkin, A.Ecer, J.Periaux, M. Satofuka and P.Fox (Editors). Elsevier B. V. 2004. P. 73–80.